



Introducción al Algoritmo Prophet de Facebook

Julio Waissman

Aprendizaje Automático Aplicado

¿Qué es Prophet?

Prophet es una herramienta de modelado de series temporales desarrollada por Facebook. Está diseñada para manejar series temporales con fuertes efectos estacionales y trabajar bien con datos faltantes y valores atípicos. Prophet está construido para ser fácil de usar y proporciona pronósticos precisos de manera rápida.

Características Principales

- **Modelado de Tendencias:** Prophet descompone la serie temporal en componentes de tendencia y estacionalidad.
- **Estacionalidad:** Puede capturar múltiples temporadas como anuales, mensuales, semanales, diarias, etc.
- **Días Festivos:** Permite incorporar el efecto de los días festivos en el pronóstico.
- **Manejo de Datos Faltantes:** Es robusto frente a datos faltantes y valores atípicos.
- **Interactividad:** Los usuarios pueden ajustar los parámetros del modelo de manera intuitiva.

Componentes del Modelo

1. **Tendencia:** Captura el crecimiento o decrecimiento general en los datos.
2. **Estacionalidad:** Modela los patrones repetitivos en los datos (por ejemplo, aumentos y disminuciones anuales).
3. **Festivos:** Permite añadir el impacto de días festivos específicos.
4. **Error:** Componentes residuales no capturados por los anteriores.

Cómo Funciona Prophet

Prophet usa una descomposición aditiva (o en algunos casos multiplicativa) de series temporales donde la serie temporal $y(t)$ se modela como:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

- $g(t)$: Modelo de tendencia que puede ser lineal o logístico.
- $s(t)$: Modelo de estacionalidad.
- $h(t)$: Efecto de días festivos.
- ϵ_t : Error aleatorio no capturado por el modelo.

Modelo de Tendencia: Crecimiento Lineal

$$g(t) = k + mt$$

donde:

- k es el nivel inicial de la serie.
- m es la tasa de cambio (pendiente) de la tendencia.

Modelo de Tendencia: Crecimiento Logístico

$$g(t) = \frac{C}{1 + \exp(-k(t - m))}$$

donde:

- C es la capacidad máxima.
- k controla la tasa de crecimiento.
- m desplaza la curva a lo largo del tiempo.

Modelo de Estacionalidad

$$s(t) = \sum_{n=1}^N \left(a_n \cos \left(\frac{2\pi n t}{P} \right) + b_n \sin \left(\frac{2\pi n t}{P} \right) \right)$$

donde:

- P es el período de la estacionalidad (por ejemplo, 365.25 para anual).
- N es el número de términos de Fourier.
- a_n y b_n son los coeficientes que se ajustan a los datos.

Festivos y Eventos Especiales

- El efecto de los días festivos $h(t)$ se modela como un incremento o decremento en el valor de la serie temporal durante los días festivos.
- Prophet permite especificar un conjunto de días festivos y ajusta un parámetro adicional para cada uno de ellos.

Error

El término de error ϵ_t captura la variabilidad no explicada por los otros componentes. Este término se asume como ruido blanco con una distribución normal de media cero:

$$\epsilon_t \sim \mathcal{N}(0, \sigma^2)$$

Implementación Práctica de Prophet

- Instalación

```
pip install fbprophet
```

- Importar Librerías

```
import pandas as pd  
from fbprophet import Prophet  
import matplotlib.pyplot as plt  
import numpy as np
```

Preparar los Datos

Los datos deben estar en un DataFrame con dos columnas: `ds` (fecha) y `y` (valor).

```
# Crear un DataFrame de ejemplo
data = {
    'ds': pd.date_range(start='2020-01-01', periods=365, freq='D'),
    'y': np.random.rand(365)
}
df = pd.DataFrame(data)
```

Entrenar el Modelo

```
# Inicializar el modelo  
model = Prophet()  
  
# Ajustar el modelo a los datos  
model.fit(df)
```

Hacer Pronósticos

Para hacer pronósticos, primero se debe crear un DataFrame que contenga las fechas futuras.

```
# Crear DataFrame con las fechas futuras  
future = model.make_future_dataframe(periods=90)  
  
# Hacer las predicciones  
forecast = model.predict(future)
```

Visualizar el Pronóstico

```
# Plotear el pronóstico  
fig = model.plot(forecast)  
plt.show()
```

Para visualizar los componentes del pronóstico (tendencia, estacionalidad, festivos):

```
# Plotear los componentes del pronóstico  
fig2 = model.plot_components(forecast)  
plt.show()
```

Uso de Variables Exógenas con Prophet

- Prophet no solo permite modelar componentes internos de una serie temporal, sino que también puede incorporar variables exógenas.
- Estas variables pueden ser cualquier tipo de datos que se cree que influyen en la serie temporal que se está modelando.
- Prophet permite agregar variables exógenas al modelo utilizando el método `add_regressor`.

Pasos para Usar Variables Exógenas

- 1. Preparar los Datos**
- 2. Agregar las Variables Exógenas al Modelo**
- 3. Entrenar el Modelo**
- 4. Hacer Pronósticos**
- 5. Visualizar los Resultados**

Preparar los Datos

Supongamos que tenemos datos de ventas diarias y también datos sobre promociones de marketing.

```
import pandas as pd
import numpy as np

# Crear datos de ejemplo
date_rng = pd.date_range(start='2020-01-01', end='2020-12-31', freq='D')
df = pd.DataFrame(date_rng, columns=['ds'])
df['y'] = np.random.rand(len(date_rng)) * 100 # Ventas diarias

# Crear datos de una variable exógena (por ejemplo, intensidad de promociones)
df['promo'] = np.random.randint(0, 2, size=(len(date_rng))) # 0 o 1 indicando si hubo promoción
```

Agregar las Variables Exógenas al Modelo y entrenar

Se debe indicar al modelo Prophet que utilice la variable exógena `promo`.

```
# Inicializar el modelo  
model = Prophet()  
  
# Añadir la variable exógena  
model.add_regressor('promo')  
  
# Ajustar el modelo a los datos  
model.fit(df)
```

Visualizar los Resultados

Podemos visualizar el pronóstico y verificar el impacto de la variable exógena.

```
import matplotlib.pyplot as plt

# Plotear el pronóstico
fig = model.plot(forecast)
plt.show()

# Plotear los componentes del pronóstico
fig2 = model.plot_components(forecast)
plt.show()
```

Consideraciones Adicionales

- **Normalización:** Asegúrate de que las variables exógenas estén normalizadas o estandarizadas si tienen escalas muy diferentes.
- **Evaluación del Modelo:** Usa métricas de evaluación como MAE, MSE, etc., para comparar modelos con y sin variables exógenas.
- **Multicolinealidad:** Ten cuidado con la multicolinealidad entre las variables exógenas.

Conclusión

- Prophet es una herramienta poderosa y fácil de usar para el pronóstico de series temporales.
- Su capacidad para manejar datos faltantes, valores atípicos y múltiples componentes de estacionalidad lo hace ideal para una variedad de aplicaciones prácticas en diferentes industrias.
- Incorporar variables exógenas en Prophet puede mejorar significativamente la precisión de los pronósticos, especialmente cuando estas variables tienen un impacto considerable en la serie temporal.

Referencias:

- [Página de recursos](#)
- [Artículo original](#)
- [Prophet en GitHub](#)
- [Documentación Oficial](#)